

FIG. 1

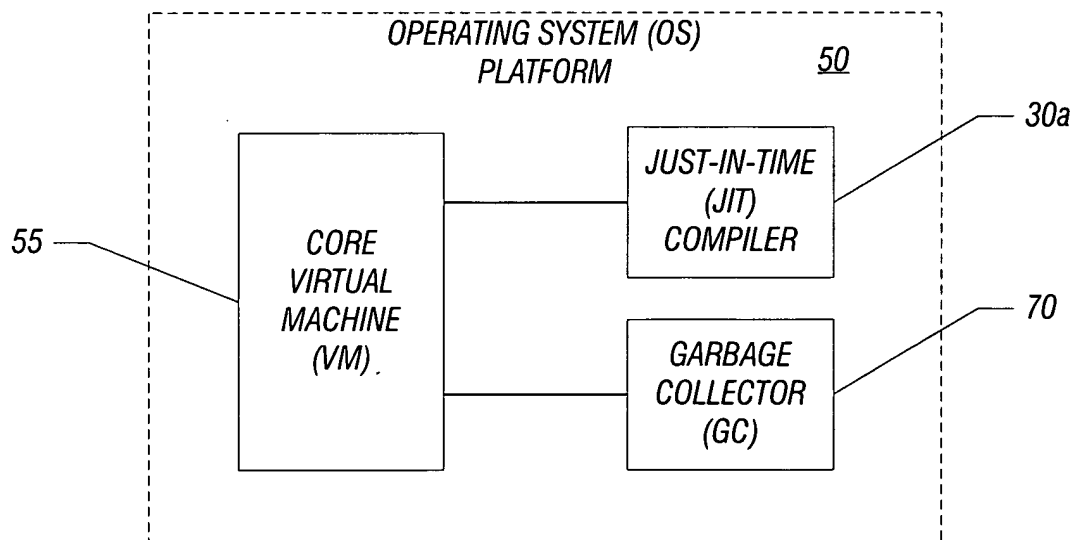


FIG. 2

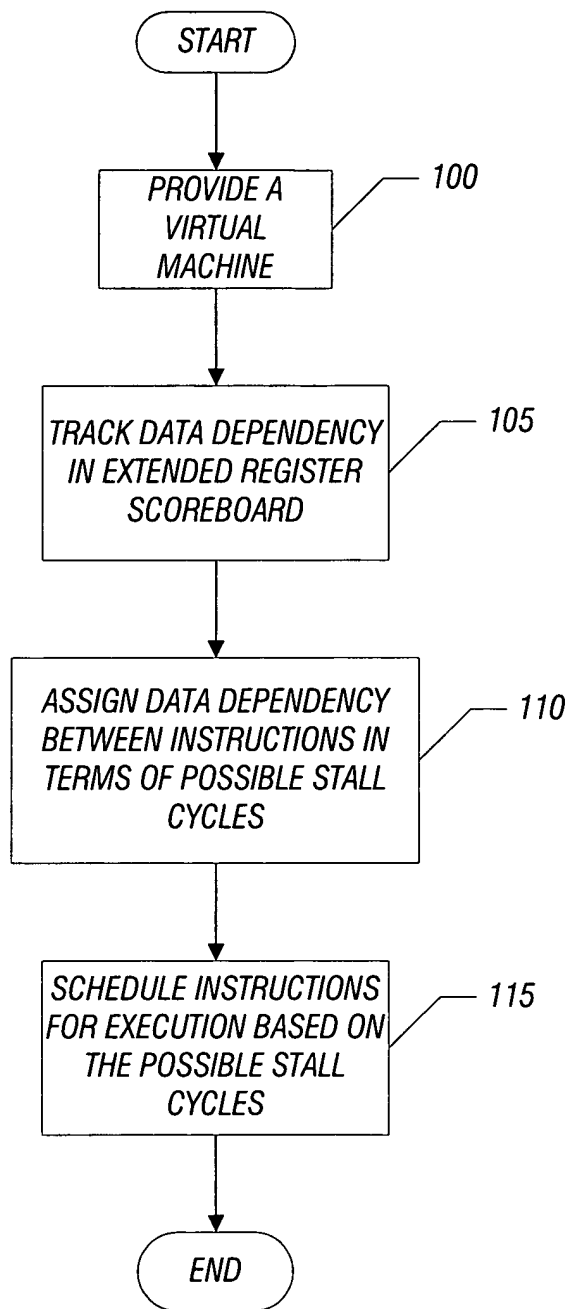


FIG. 3

I0: mov r0, r1
I1: mov r2, 0x3
I2: mov r3, 0x63
I3: add r12, r1, 0x8
I4: ldr lr, [r1, 0x4]
I5: cmp lr, r2

125

FIG. 4

	GAP	G_C	UP	DWN	L	L0	I0	I1	I2	I3	I4	I5
I0	0	\	-1	6	1	1	\					
I1	0	\	-1	5	2	1	-1	\				
I2	0	\	-1	6	3	1	-1	-1	\			
I3	0	\	-1	6	4	1	-1	-1	-1	\		
I4	0	\	-1	5	5	1	-1	-1	-1	-1	\	
I5	2	4	4	6	7	1	-1	0	-1	-1	2	\

FIG. 5

```

1.  for (every GAP(n) > 0 ){
2.      for(m = index(G_C) - 1 ; m >= 0 && GAP(n) > 0; m--){
3.          if( m is in another GAP) break ;
4.          if( m has been moved) break ;
5.          if( DWN(m) > index(G_C) ){
6.              move m before index(G_C)+1 ;
7.              GAP(n) = GAP(n) - L0(m) ;
8.          }
9.      }
10.
11.     for(m = n + 1 ; m <= last instruction && GAP(n) > 0; m++){
12.         if( m is in another GAP) break ;
13.         if( m has been moved) break ;
14.         if( UP(m) < n ){
15.             move m after n-1 ;
16.             GAP(n) = GAP(n) - L0(m) ;
17.         }
18.     }
19. }
```

FIG. 6

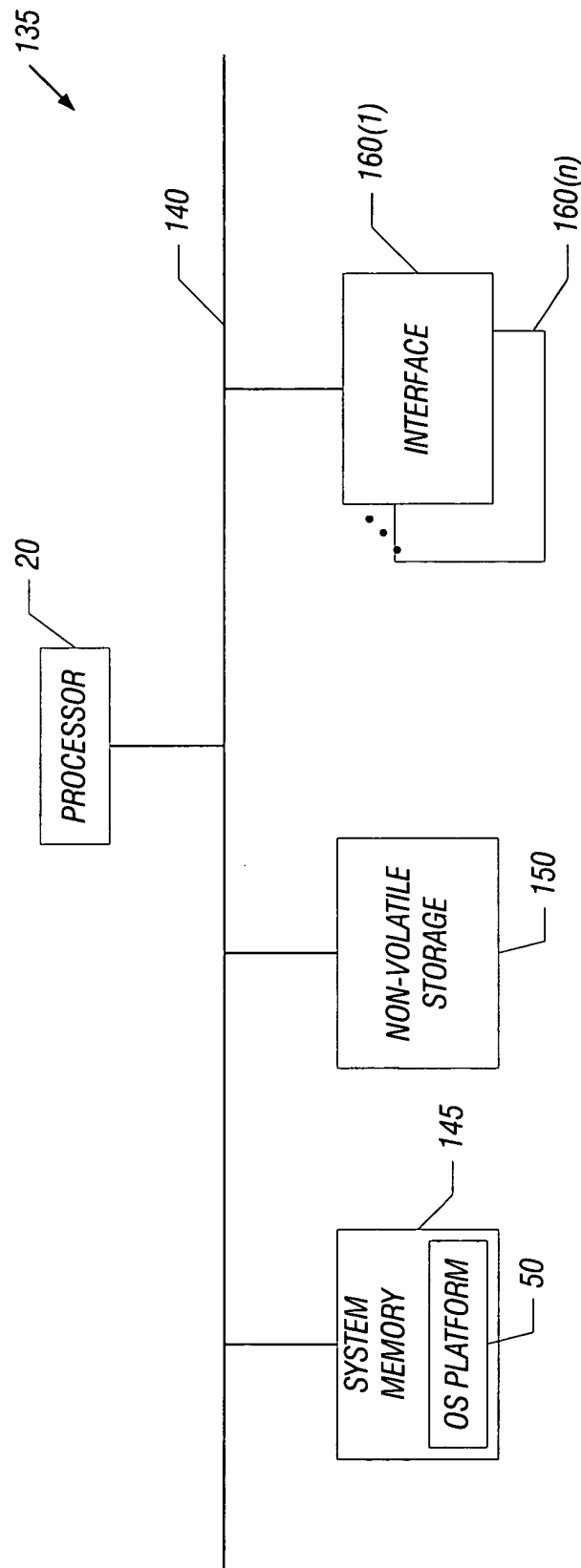


FIG. 7